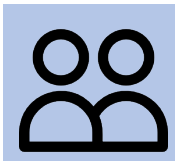
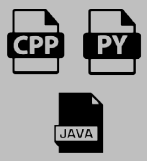
Taller en Sala Nro. 4
Notación O Recursiva

En la vida real, la serie de Fibonacci se utiliza para diseñar la distribución de objetos en los videojuegos de tal forma que las escenas cumplan con una cierta estética determinada por la proporción aurea de Fibonacci. Así: http://bit.ly/2hIQQe1

 **Trabajo en Parejas**	 **Hoy, plazo máximo de entrega**	 **Docente entrega código suelto en GitHub**
 **Sí .cpp, .py o .java**	 **No .zip, .txt, html o .doc**	 **Alumnos entregan código suelto por GitHub**

Ejercicios a resolver

Consideren lo siguientes algoritmos:

1. ArrayMax, calcula el máximo valor de un arreglo de enteros. A es el arreglo y n es la última posición del arreglo en el primer llamado. *ArrayMax* se llama recursivamente hasta que n sea igual a 0.

 **Pista 1:** Como un ejemplo ArrayMax([1,2,3,8,10,4], 5) retorna 10.

 **Pista 2:** El tamaño del problema aquí es el número de elementos del arreglo

DOCENTE MAURICIO TORO BERMÚDEZ
Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627
Correo: mtorobe@eafit.edu.co

```
SubProceso ArrayMax( A, n )
      Definir i, max, temp Como Entero;
             max <- A[n]; // Si n = 0, max <- A[0]
             Si n != 0 Entonces
                   temp <- ArrayMax(A, n-1);
                   Si temp > max Entonces
                       max <- temp;
      Retornar max;
```

2. El algoritmo **groupSum**, dado un arreglo de enteros, decide si es posible escoger un subconjunto de esos enteros, de tal forma que la suma de los elementos de ese subconjunto sea igual al parámetro ***target***. El parámetro ***start*** funciona como un contador y representa un índice en el arreglo de números ***nums***.

Pista 1: El tamaño del problema aquí es el número de elementos del arreglo

```
public boolean groupSum(int start, int[] nums, int target) {
    if (start >= nums.length) return target == 0;
    return groupSum(start + 1, nums, target - nums[start])
            || groupSum(start + 1, nums, target);
}
```

3. El algoritmo fibonacci calcula el valor enésimo de la serie de Fibonacci recursivamente.

Pista 1: El tamaño del problema es el término enésimo (int n).

```
    public static long fibonacci(int n) {
        if (n <= 1) return n;
        else return fibonacci(n-1) + fibonacci(n-2);
    }
```

DOCENTE MAURICIO TORO BERMÚDEZ
Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627
Correo: mtorobe@eafit.edu.co

Teniendo en cuenta lo anterior:

1. Implementen el algoritmo de *ArrayMax* en Java y copiar los otros dos
2. Identifiquen qué representa el tamaño del problema para cada algoritmo
3. Identifiquen valores apropiados para tamaños del problema
4. Tomen los tiempos para los anteriores algoritmos con 20 tamaños del problema diferentes.
5. Hacer una gráfica en Excel para cada algoritmo y pasarla a Word luego
6. Entregar el archivo de Word pasado a pdf. No entregar el código.

DOCENTE MAURICIO TORO BERMÚDEZ
Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627
Correo: mtorobe@eafit.edu.co

Ayudas para resolver los ejercicios

Ayudas para los ejercicios

Pista 1: Vean la sección 4.2 de la Guía de Laboratorios muestra cómo generar arreglos con números aleatorios

Pista 2: Vean la sección 4.3 de la Guía de Laboratorios muestra cómo aumentar el *heap* en Java y la sección 4.4 muestra cómo aumentar el tamaño de la pila.

Pista 3: Vean la sección 4.7 de la Guía de Laboratorios muestra cómo tomar el tiempo en Java

Pista 4: Vean la sección 4.6 de la Guía de Laboratorios muestra cómo cambiar la escala de una gráfica en Excel a escala logarítmica.

DOCENTE MAURICIO TORO BERMÚDEZ
Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627
Correo: mtorobe@eafit.edu.co

¿Alguna inquietud?

CONTACTO

Docente Mauricio Toro Bermúdez
Teléfono: (+57) (4) 261 95 00 **Ext.** 9473
Correo: mtorobe@eafit.edu.co
Oficina: 19- 627

Agende una cita con él a través de **http://bit.ly/2gzVg10** , en la pestaña *Semana. Si no da clic en esta pestaña, parecerá que toda la agenda estará ocupada.*